

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-312324

(43)Date of publication of application : 09.11.2001

(51)Int.Cl.

G06F 1/00 G06F 13/00

G06F 15/00 G06F 17/60

(21)Application number : 2000-131796

(71)Applicant : NEC NEXSOLUTIONS LTD

(22)Date of filing : 28.04.2000

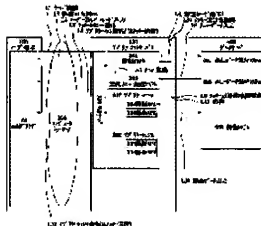
(72)Inventor : MORITA MASAO

(54) METHOD FOR AUTHENTICATING AND CHARGING USER, RECORDING MEDIUM THEREFOR, METHOD AND SYSTEM FOR PROVIDING APPLICATION SERVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method and a system for providing an application service, with which a vender can be easily participated in the business of an application service provider.

SOLUTION: A previously authenticated certificate is inputted to an application program(AP), the previously authenticated certificate is inputted through an authenticating API to an authenticating system, a certificate authenticated again is outputted through the authenticating API to the AP, a conversation request to the AP is inputted to the AP, a conversation response to the conversation request is prepared by the AP, the result of utilization corresponding to the conversation request is prepared by the AP and the result of utilization is inputted through a charging API to a charging



system.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-312324

(P2001-312324A)

(43) 公開日 平成13年11月9日 (2001.11.9)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)	
G 0 6 F 1/00		G 0 6 F 13/00	5 3 0 S	5 B 0 4 9
13/00	5 3 0	15/00	3 3 0 B	5 B 0 7 6
15/00	3 3 0	17/00	3 3 2	5 B 0 8 5
17/00	3 3 2	9/06	6 6 0 A	

審査請求 有 請求項の数20 O L (全 24 頁)

(21) 出願番号 特願2000-131796 (P2000-131796)

(22) 出願日 平成12年4月28日 (2000.4.28)

(71) 出願人 390001041

エヌイーシーネクスソリューションズ株式
会社
東京都港区三田1丁目4番28号

(72) 発明者 森田 将夫

東京都港区三田1丁目4番28号 日本電気
情報サービス株式会社内

(74) 代理人 100095740

弁理士 開口 宗昭

Fターム (参考) 5B049 A02

5B076 FA20 FB05

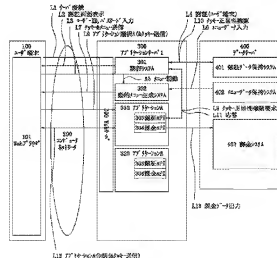
5B085 A08 A04 AE02 AE23

(54) 【発明の名称】 ユーザの認証及び課金方法、その記録媒体、アプリケーション・サービスを提供する方法及びシステム

(57) 【要約】

【課題】 ベンダーがアプリケーション・サービス・プログラマの事業に容易に参加できるアプリケーション・サービスを提供する方法及びシステムを提供することを課題とする。

【解決手段】 予め認証された認証証明書が、アプリケーションプログラムに入力され、予め認証された認証証明書が、認証APIを介して認証システムに入力され、再度認証された認証証明書が、認証APIを介してアプリケーションプログラムに出力され、アプリケーションプログラムに対する会話要求が、アプリケーションプログラムに入力され、会話要求に対する会話応答が、アプリケーションプログラムによって作成され、会話要求に対する利用実績が、アプリケーションプログラムによって作られ、利用実績が、課金APIを介して課金システムに入力されることによって上記課題を解決する。



【特許請求の範囲】

【請求項1】 認証システムによって予め認証された認証証明書が、アプリケーションプログラムに入力され、前記予め認証された認証証明書が、認証API(Application Program Interface)を介して前記認証システムに入力され、且つ前記認証システムによって再度認証された認証証明書が、前記認証APIを介して前記アプリケーションプログラムに出力されるステップを含むことと特徴とする認証関数プログラムによるユーザの認証方法。

【請求項2】 請求項1に記載のステップを実行させるための認証関数プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項3】 アプリケーションプログラムに対する会話要求が、前記アプリケーションプログラムに入力される第1のステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第2のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第3のステップと、前記利用実績が、課金APIを介して課金システムに入力される第4のステップと、を含むことと特徴とする課金関数プログラムによるユーザの課金方法。

【請求項4】 前記第1のステップ後前記第2のステップ前に処理開始時刻が、前記アプリケーションプログラムによって作成されるステップと、前記第2のステップ後前記第3のステップ前に処理終了時刻が、前記アプリケーションプログラムによって作成されるステップと、を含む、前記第3のステップは、前記処理開始時刻及び前記処理終了時刻に基づいて前記利用実績が作成されることを特徴とする請求項3に記載の課金関数プログラムによるユーザの課金方法。

【請求項5】 前記第3のステップは、前記会話要求及び前記会話応答に基づく課金項目に基づき前記利用実績が作成されることを特徴とする請求項3又は請求項4に記載の課金関数プログラムによるユーザの課金方法。

【請求項6】 請求項3から請求項5の何れかに記載の全てのステップを実行させるための課金関数プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項7】 認証システムによって予め認証された認証証明書が、ユーザ端末からアプリケーションプログラムに受信され、前記予め認証された認証証明書が、前記認証APIを介して前記認証システムに入力され、且つ前記認証システムによって再度認証された認証証明書が、前記認証APIを介して前記アプリケーションプログラムからユーザ端末に送信されるステップを含むことと特徴とするアプリケーションプログラムによるユーザの認証方法。

【請求項8】 アプリケーションプログラムに対する会話要求が、ユーザ端末から前記アプリケーションプログラムに受信される第1のステップと、前記会話要求に対

する会話応答が、前記アプリケーションプログラムによって作成される第2のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第3のステップと、前記利用実績が、課金APIを介して前記アプリケーションプログラムから課金システムに入力される第4のステップと、を含むことと特徴とするアプリケーションプログラムによるユーザの課金方法。

【請求項9】 前記第1のステップ後前記第2のステップ前に処理開始時刻が、前記アプリケーションプログラムによって作成されるステップと、前記第2のステップ後前記第3のステップ前に処理終了時刻が、前記アプリケーションプログラムによって作成されるステップと、を含む、前記第3のステップは、前記処理開始時刻及び前記処理終了時刻に基づいて前記利用実績が作成されることを特徴とする請求項8に記載のアプリケーションプログラムによるユーザの課金方法。

【請求項10】 前記第3のステップは、前記会話要求及び前記会話応答に基づく課金項目に基づき前記利用実績が作成されることを特徴とする請求項8又は請求項9に記載のアプリケーションプログラムによるユーザの課金方法。

【請求項11】 請求項7に記載のステップと、請求項8から請求項10の何れかに記載の全てのステップと、を含むことを特徴とするユーザの課金及び課金方法。

【請求項12】 請求項7から請求項11の何れかに記載の全てのステップを実行させるためのアプリケーションプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項13】 ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに受信される第1のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証APIを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、前記認証APIを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第2のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第3のステップと、前記再度認証された認証証明書と前記会話

応答とか、前記アプリケーションプログラムからユーザ端末に送信されるステップと、前記利用実績が、課金APIを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第4のステップと、を含むことと特徴とするアプリケーションサーバによるユーザの認証及び課金方法。

【請求項14】 請求項13に記載の全てのステップを実行させるためのアプリケーションサーバに格納されるアプリケーションプログラム及び認証システムプログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項15】 ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに送受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに送受信される第1のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証APIを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、認証APIを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第2のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第3のステップと、前記再度認証された認証証明書と前記会話応答とが、前記アプリケーションプログラムからユーザ端末に送受信されるステップと、前記利用実績が、課金APIを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第4のステップと、を含むことと特徴とするアプリケーション・サービスを提供する方法。

【請求項16】 前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納された認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな認証証明書と認証保持システムによって変更され、且つ前記新たな認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな認証証明書として出力されることを特徴とする請求項15に記載のアプリケーション・サービスを提供する方法。

【請求項17】 前記認証ステップに係る認証証明書が、前記認証システムによって予め暗号化された暗号化済認証証明書であり、前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納さ

れた認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな暗号化済認証証明書と前記認証システムによって暗号化され、且つ前記新たな暗号化済認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな暗号化済認証証明書として出力されることを特徴とする請求項15又は請求項16に記載のアプリケーション・サービスを提供する方法。

【請求項18】 ユーザ識別をユーザ端末から受信し、前記ユーザ識別を認証して予め認証された認証証明書がユーザ端末に送信し、前記予め認証された認証証明書をアプリケーションプログラムから前記アプリケーションプログラムに組み込まれた認証APIを介して入力され、前記予め認証された認証証明書を再度認証された認証証明書を前記アプリケーションプログラムに前記認証APIを介して出力する認証システムと、前記会話要求と前記予め認証された認証証明書とを前記ユーザ端末から受信し、前記予め認証された認証証明書を前記認証システムに入力し、前記再度認証された認証証明書を前記認証システムから出力され、前記会話要求に対する会話応答と利用実績とを作成し、前記前記再度認証された認証証明書と前記会話応答とを前記ユーザ端末に送信し、前記利用実績を課金APIを介してサーバに格納される課金システムに入力するアプリケーションプログラムと、備えることを特徴とするアプリケーションサーバ。

【請求項19】 アプリケーションサーバに格納される認証システムによって予め認証された認証証明書を前記認証システムから格納され、前記認証システムによって再度認証された認証証明書を前記認証システムから格納される認証保持システムと、利用実績を前記アプリケーションサーバに格納されるアプリケーションから前記アプリケーションプログラムに組み込まれた認証APIを介して入力される課金システムと、を備えるサーバ。

【請求項20】 請求項18に記載のアプリケーションサーバと、請求項19に記載のサーバと、から構成されるアプリケーション・サービスを提供するシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ユーザの認証及び課金方法、その記録媒体、アプリケーション・サービスを提供する方法及びシステムに関するものである。

【0002】

【発明の背景】 アプリケーション・サービスを提供するとは、コンピュータネットワークを介してアプリケーションプログラムを有償でユーザに利用させること、若しくはコンピュータネットワークを介していわゆる商用サービスをユーザに提供することという。

【0003】ここで、コンピュータネットワークとは、例えば、インターネット(Internet)、広域通信網(WAN: Wide Area Network)等である。インターネットとは、良く知られるように、通信プロトコルTCP/IP (Transmission Control Protocol/Internet Protocol)を用いて全世界のネットワークを相互に接続した巨大なコンピュータネットワークである。また、TCP/IPとはインターネット、インターネットで標準的に使われるプロトコルであり、TCP/IPはHTTP (Hypertext Transfer Protocol)、FTP (File Transfer Protocol)等の基盤となるプロトコルである。尚、HTTPとはWebサーバとWebブラウザとの間でHTML文書、画像ファイル、音楽ファイル、映像ファイル等を送受信するのに使われるプロトコルであり、HTTPは1つのHTML文書毎にTCP/IPの接続・転送(HTML文書のほか、このHTML文書に関連付けられている画像ファイル等)・切断を行うステータスのプロトコルである。HTTPは、IETF(Internet Engineering Task Force)が公開している技術文書RFC(Request for Comments)2616で定義されている(<http://www.ietf.org/rfc/rfc2616.txt>)。

【0004】また、WebサーバとはWebブラウザからのリクエストに応じてサーバ上のリソース(HTML文書、画像ファイル、音楽ファイル、映像ファイル等)をWebブラウザに送信するものである。ここで、Webサーバには、サーバ上のHTML文書等を単に送信するものほか、サーバ上のアプリケーションプログラムを実行しその出力結果に基づいて生成されたHTML文書等を送信するものを含む。尚、アプリケーションプログラムの出力結果に基づいてHTML文書を生成するアプリケーションプログラムは、例えば、Perl(Practical Extraction and Report Language)、Java(登録商標)等のプログラミング言語で作成されている。Perlで作成されたアプリケーションプログラムは、良く知られるように、CGI(Common Gateway Interface)技術を用いて動的なHTML文書を生成できる。一方、Webブラウザとはサーバ上のリソースをWebブラウザから受信したリソースを解析しユーザ端末のディスプレイ又はスピーカに表示又は再生するものである。加えて、Webブラウザはユーザからの入力を受け取ってWebサーバに送信することもある。ここで、Webブラウザには、Netscape Communications社のNetscape Navigator(Communicator)、Microsoft社のInternet Explorer等がある。

【0005】ところで、現在、アプリケーション・サービスを提供する事業者、即ちアプリケーション・サービス・プロバイダ(ASP: Application Service Provider)と言え、自らアプリケーションプログラムを開発するベンダーである。即ち、ベンダー自らがアプリケーションプログラムを開発するとともに、業務プロセス

プログラム(ユーザ認証プログラム、課金処理プログラム等)を開発して、業務プロセスプログラムを組み込んだアプリケーションプログラムをコンピュータネットワークを介して有償でユーザに利用させている。

【発明が解決しようとする課題】現在のアプリケーション・サービスを提供する方法及びシステムには次のような問題があった。

【0006】ベンダー自ら業務プロセスプログラム(ユーザ認証プログラム、課金処理プログラム等)を開発する必要があるため、業務プロセスプログラムの開発に必要な設備(プログラム開発者、プログラム開発用サーバ等)を導入する必要がある。即ち、ベンダー自らが業務プロセスプログラムの開発に必要な設備を導入し、業務プロセスを運用する必要がある。この設備の導入コスト及び業務プロセスの運用コストは高額であるため、事実上多くの資本を有する大手ベンダーのみがアプリケーション・サービス・プロバイダ(ASP)となっている。従って、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できないという問題があった。

【0007】また、業務プロセスプログラムの開発に必要な設備の導入コスト及び業務プロセスの運用コストは高額であるため、大手ベンダーでさえアプリケーションプログラムを高価格でユーザに利用させて、初めて導入コスト及び運用コストの採算が取れているのが現状である。従って、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が事実上少ないという問題があった。

【0008】更に、大手ベンダーが個々にアプリケーション・サービス・プロバイダ(ASP)となるため、ユーザはアプリケーション・サービス・プロバイダ(ASP)毎に予め契約を結び、且つアプリケーションプログラムの利用代金を支払う必要がある。従って、ユーザは契約、利用代金の支払い等の事務手が煩雑であるという問題があった。

【0009】本発明は現在のアプリケーション・サービスを提供する方法及びシステムにおける問題に鑑みてなされたものであって、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるユーザの認証及び課金方法、その記録媒体、アプリケーション・サービスを提供する方法及びシステムを提供することを課題とする。

【課題を解決するための手段】前記課題を解決する請求項1は、認証システムによって予め認証された認証証明書が、アプリケーションプログラムに入力され、前記予め認証された認証証明書が、認証API(Application Program Interface)を介して前記認証システムに入力され、且つ前記認証システムによって再度認証された認証証明書が、前記認証APIを介して前記アプリケーションプログラムに出力されるステップを含むことと特徴とす

る認証関数プログラムによるユーザの認証方法である。
 【0010】また請求項2は、請求項1に記載のステップ
 を実行させるための認証関数プログラムを記録したコ
 ンピュータ読み取り可能な記録媒体である。

【0011】したがって請求項1又は請求項2の発明によ
 れば、アプリケーションプログラムに認証API (App
 lication Program Interface) を組み込むだけで、アプ
 リケーションプログラムはユーザの管理ができる。即
 ち、アプリケーションプログラムを開発するだけで、ユ
 ーザ認証プログラムを開発する必要がなくなる。これに
 より、ベンダーがアプリケーション・サービス・プロバ
 イダの事業に参入できるという利点がある。

【0012】また請求項3は、アプリケーションプログラ
 ムに対する会話要求が、前記アプリケーションプログラ
 ムに入力される第1のステップと、前記会話要求に対
 する会話応答が、前記アプリケーションプログラムによ
 って作成される第2のステップと、前記会話要求に対
 する利用実績が、前記アプリケーションプログラムによ
 って作成される第3のステップと、前記利用実績が、課金
 API を介して課金システムに入力される第4のステッ
 プと、を含むことと特徴とする課金関数プログラムによる
 ユーザの課金方法である。

【0013】したがって請求項3の発明によれば、アプ
 リケーションプログラムに課金API (Application Pro
 gram Interface) を組み込むだけで、アプリケーション
 プログラムはユーザの課金ができる。即ち、アプリー
 ションプログラムを開発することだけで、課金処理プロ
 グラムを開発する必要がなくなる。これにより、ベン
 ダーがアプリケーション・サービス・プロバイダの事
 業に参入できるという利点がある。

【0014】また請求項4は、前記第1のステップ後前
 記第2のステップ前に処理開始時刻が、前記アプリー
 ションプログラムによって作成されるステップと、前記
 第2のステップ後前記第3のステップ前に処理終了時刻
 が、前記アプリケーションプログラムによって作成され
 るステップと、を含み、前記第3のステップは、前記処
 理開始時刻及び前記処理終了時刻に基づいて前記利用実
 績が作成されることを特徴とする請求項3に記載の課金
 関数プログラムによるユーザの課金方法である。

【0015】また請求項5は、前記第3のステップは、
 前記会話要求及び前記会話応答に基づく課金項目に基
 づき前記利用実績が作成されることを特徴とする請求
 項3又は請求項4に記載の課金関数プログラムによるユ
 ーザの課金方法である。

【0016】また請求項6は、請求項3から請求項5の
 何れか一に記載の全てのステップを実行させるための課
 金プログラムを記録したコンピュータ読み取り可能な
 記録媒体である。

【0017】したがって請求項4から請求項6の何れか
 一の発明によれば、アプリケーションプログラムはユー

ザの利用態様に応じて課金ができる。

【0018】また請求項7は、認証システムによって予
 め認証された認証証明書が、ユーザ端末からアプリー
 ションプログラムに受信され、前記予め認証された認
 証証明書が、前記認証API を介して前記認証システ
 ムに入力され、且つ前記認証システムによって再度認
 証された認証証明書が、前記認証API を介して前記ア
 プリケーションプログラムからユーザ端末に送信され
 るステップとを含むことと特徴とするアプリケーション
 プログラムによるユーザの認証方法である。

【0019】また請求項8は、アプリケーションプログラ
 ムに対する会話要求が、ユーザ端末から前記アプリー
 ションプログラムに受信される第1のステップと、前
 記会話要求に対する会話応答が、前記アプリケーション
 プログラムによって作成される第2のステップと、前
 記会話要求に対する利用実績が、前記アプリケーション
 プログラムによって作成される第3のステップと、前記
 利用実績が、課金API を介して前記アプリケーション
 プログラムから課金システムに入力される第4のステッ
 プと、を含むことと特徴とするアプリケーションプログラ
 ムによるユーザの課金方法である。

【0020】また請求項9は、前記第1のステップ後前
 記第2のステップ前に処理開始時刻が、前記アプリー
 ションプログラムによって作成されるステップと、前記
 第2のステップ後前記第3のステップ前に処理終了時刻
 が、前記アプリケーションプログラムによって作成され
 るステップと、を含み、前記第3のステップは、前記処
 理開始時刻及び前記処理終了時刻に基づいて前記利用実
 績が作成されることを特徴とする請求項8に記載のア
 プリケーションプログラムによるユーザの課金方法であ
 る。

【0021】また請求項10は、前記第3のステップ
 は、前記会話要求及び前記会話応答に基づく課金項目に
 基づき前記利用実績が作成されることを特徴とする請
 求項8又は請求項9に記載のアプリケーションプログラ
 ムによるユーザの課金方法である。

【0022】また請求項11は、請求項7に記載のステ
 ップと、請求項8から請求項10の何れか一に記載の全
 てのステップと、を含むことを特徴とするユーザの課
 金及び課金方法である。

【0023】また請求項12は、請求項7から請求項1
 1の何れか一に記載の全てのステップを実行させるた
 めのアプリケーションプログラムを記録したコンピ
 ュータ読み取り可能な記録媒体である。

【0024】したがって請求項7から請求項12の何れ
 か一の発明によれば、認証API 及び/又は課金API
 を組み込んだアプリケーションプログラムを作成する
 だけで、アプリケーションプログラムはユーザの認証及
 び/又は課金ができる。即ち、認証API 及び/又は課
 金API を組み込んだアプリケーションプログラムを開発

するだけで、ユーザ認証及び/又は課金処理プログラムを開発する必要がなくなる。これにより、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【0025】また請求項13は、ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに受信される第1のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証APIを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、認証APIを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第2のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第3のステップと、前記再度認証された認証証明書と前記会話応答とが、前記アプリケーションプログラムからユーザ端末に送信されるステップと、前記利用実績が、課金APIを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第4のステップと、を含むこと特徴とするアプリケーションサーバによるユーザの認証及び課金方法である。

【0026】また請求項14は、請求項13に記載の全てのステップを実行させるためのアプリケーションサーバに格納されるアプリケーションプログラム及び認証システムプログラムを記録したコンピュータ読み取り可能な記録媒体である。

【0027】したがって請求項13又は請求項14の発明によれば、認証API及び課金APIを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンダーはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【0028】また請求項15は、ユーザ識別が、ユーザ端末からアプリケーションサーバに格納される認証システムに受信されるログインステップと、前記ユーザ識別が、前記認証システムによって認証され、前記認証システムによって予め認証された認証証明書が、前記認証システムから前記ユーザ端末に送信される認証ステップと、アプリケーションプログラムに対する会話要求と前

記予め認証された認証証明書とが、前記ユーザ端末から前記アプリケーションプログラムに受信される第1のステップと、前記予め認証された認証証明書が、前記アプリケーションプログラムに組み込まれた認証APIを介して前記認証システムに入力され、前記予め認証された認証証明書が、前記認証システムによって再度認証され、前記認証システムによって再度認証された認証証明書が、認証APIを介して前記アプリケーションプログラムに出力される再認証ステップと、前記会話要求に対する会話応答が、前記アプリケーションプログラムによって作成される第2のステップと、前記会話要求に対する利用実績が、前記アプリケーションプログラムによって作成される第3のステップと、前記再度認証された認証証明書と前記会話応答とが、前記アプリケーションプログラムからユーザ端末に送信されるステップと、前記利用実績が、課金APIを介して前記アプリケーションプログラムからサーバに格納される課金システムに入力される第4のステップと、を含むこと特徴とするアプリケーション・サービスを提供する方法である。

【0029】したがって請求項15の発明によれば、認証API及び課金APIを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンダーはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという利点がある。

【0030】また、ベンダーはユーザ認証及び課金処理プログラムを開発する必要がなくなるので、新たなアプリケーションプログラムを開発することができる。この場合、この新たなアプリケーションプログラムをアプリケーションサーバに格納するだけで、ベンダーは複数のアプリケーション・サービスを提供することができる。即ち、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が多くなるという利点がある。ここで、複数のアプリケーションプログラムがそれぞれ認証APIを介して一の認証システムによって、ユーザを識別することができる。換言すれば、複数のアプリケーションプログラム間でユーザ管理の統一性を実現することができる。

【0031】更に、別のベンダーが別のアプリケーションプログラムを開発した場合、複数のアプリケーションプログラムと一の認証システムとをアプリケーションサーバに別のアプリケーションプログラムを格納するだけで、別のベンダーもアプリケーション・サービスを提供することができる。即ち、アプリケーションサーバは複数のベンダーのアプリケーションプログラムをユーザに有償で利用させることができる。ここで、複数のベンダーのアプリケーションプログラムがそれぞれ課金APIを介して一の課金システムによって、利用実績を集計して利用金額を算出することができる。換言すれば、複数

のベンダーの間で課金管理の統一性を実現することができる。課金システムはユーザ別に利用金額を算出してユーザに支払いを求める一方、課金システムはベンダー別に利用金額を算出してベンダーに支払う。これにより、ユーザは利用代金の支払い等の事務手続が煩雑でないという利点がある。

【0032】また請求項16は、前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納された認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな認証証明書に前記認証システムによって変更され、且つ前記新たな認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな認証証明書として出力されることを特徴とする請求項15に記載のアプリケーション・サービスを提供する方法である。

【0033】また請求項17は、前記認証ステップに係る認証証明書が、前記認証システムによって予め暗号化された暗号化済認証証明書であり、前記再認証ステップは、前記予め認証された認証証明書と認証保持システムに予め格納された認証証明書とが前記認証システムによって比較されて一致する場合に、前記予め格納される認証証明書が、新たな暗号化済認証証明書に前記認証システムによって暗号化され、且つ前記新たな暗号化済認証証明書が、前記サーバに格納される認証保持システムに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな暗号化済認証証明書として出力されることを特徴とする請求項15又は請求項16に記載のアプリケーション・サービスを提供する方法である。

【0034】したがって請求項16又は請求項17の発明によれば、認証証明書の偽造を防止することができるという利点がある。

【0035】また請求項18は、ユーザ識別をユーザ端末から受信し、前記ユーザ識別を認証して予め認証された認証証明書ユーザ端末に送信し、前記予め認証された認証証明書をアプリケーションプログラムから前記アプリケーションプログラムに組み込まれた認証APIを介して入力され、前記予め認証された認証証明書を再度認証して再度認証された認証証明書を前記アプリケーションプログラムに前記認証APIを介して出力する認証システムと、前記会話要求と前記予め認証された認証証明書とを前記ユーザ端末から受信し、前記予め認証された認証証明書を前記認証システムに入力し、前記再度認証された認証証明書を前記認証システムから出力され、前記会話要求に対する会話応答と利用実績とを作成し、前記前記再度認証された認証証明書と前記会話応答とを前記ユーザ端末に送信し、前記利用実績を課金APIを介してサーバに格納される課金システムに入力するアプリケーションプログラムと、備えることを特徴とするア

プリケーションサーバである。

【0036】また請求項19は、アプリケーションサーバに格納される認証システムによって予め認証された認証証明書を前記認証システムから格納され、前記認証システムによって再度認証された認証証明書を前記認証システムから格納される認証保持システムと、利用実績を前記アプリケーションサーバに格納されるアプリケーションから前記アプリケーションプログラムに組み込まれた認証APIを介して入力される課金システムと、を備えるサーバである。

【0037】また請求項20は、請求項18に記載のアプリケーションサーバと、請求項19に記載のサーバと、から構成されてなるアプリケーション・サービスを提供するシステムである。

【0038】したがって請求項18から請求項20の何れかの発明によれば、認証API及び課金APIを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンダーはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス、プログラムの事業に容易に参入できるという利点がある。

【0039】また、ベンダーはユーザ認証及び課金処理プログラムを開発する必要がなくなるので、新たなアプリケーションプログラムを開発することができる。この場合、この新たなアプリケーションプログラムをアプリケーションサーバに格納するだけで、ベンダーは複数のアプリケーション・サービスを提供することができる。即ち、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が多くなるという利点がある。ここで、複数のアプリケーションプログラムがそれぞれ認証APIを介して一の認証システムによって、ユーザを識別することができる。換言すれば、複数のアプリケーションプログラム間でユーザ管理の統一性を実現することができる。

【0040】更に、別のベンダーが別のアプリケーションプログラムを開発した場合、複数のアプリケーションプログラムと一の認証システムとをアプリケーションサーバに別のアプリケーションプログラムを格納するだけで、別のベンダーもアプリケーション・サービスを提供することができる。即ち、アプリケーションサーバは複数のベンダーのアプリケーションプログラムによって、利用実績を有償で利用させることができる。ここで、複数のベンダーのアプリケーションプログラムがそれぞれ課金APIを介して一の課金システムによって、利用実績を累計して利用金額を算出することができる。換言すれば、複数のベンダーの間で課金管理の統一性を実現することができる。課金システムはユーザ別に利用金額を算出してユーザに支払いを求める一方、課金システムはベンダー別に利用金額を算出してベンダーに支払う。これにより、ユーザは利用代金の支払い等の事務手続が煩雑でないとい

いう利点がある。

【0041】尚、現在のアプリケーション・サービスを提供する方法及びシステムには次のような問題もあった。

【0042】例えば、複数のユーザが同一のアプリケーションプログラムを同時に利用することも当然に生じ得る。この場合、アプリケーションプログラムは複数のユーザのそれぞれに対するHTML文書等のを生成する必要があるので、アプリケーションプログラムの処理速度が低下することになる。

【0043】ここで、アプリケーションプログラムの処理速度の低下を防ぐためには、アプリケーションプログラムが格納されるサーバ（以下、「アプリケーションサーバ」とも言う。）の処理能力を高くすれば解決する。ところが、アプリケーションサーバのコストは処理能力が高くなるに伴って飛躍的に高くなるのが現状である。具体的に言えば、アプリケーションサーバのコストは、演算速度に依存するCPU(Central Processing Unit)のクロック周波数、の送信速度に依存するルータの通信速度等が2倍になると2倍以上になることは言うまでもない。また、現在の技術ではCPUのクロック周波数等には上限があるのが現状である。従って、アプリケーションサーバの処理能力を高くすることには限界があるので、ある程度の数のユーザが同一のアプリケーションプログラムを同時に利用する場合には、アプリケーションプログラムの処理速度の低下を防ぐことができるが、非常に多くの数のユーザが同一のアプリケーションプログラムを同時に利用する場合、アプリケーションプログラムの処理速度が低下することになる。

【0044】また、アプリケーションプログラムの処理速度の低下を防ぐためには、アプリケーションサーバを複数台設置することによっても解決する。即ち、アプリケーションプログラムを複数のアプリケーションサーバに分散させて、一台のアプリケーションサーバに格納されるアプリケーションプログラムを利用するユーザの数を分散させる。これにより、アプリケーションプログラムの処理速度の低下を防止することができる。結果として、例えば、同一のユーザが同一のアプリケーションプログラムを複数のアプリケーションサーバで利用することも当然に生じ得る。即ち、同一のユーザが同一のアプリケーションプログラムをあるアプリケーションサーバで利用し始め、利用を一旦中止する（一時退避）。しかる後に同一のユーザが同一のアプリケーションプログラムを別のアプリケーションサーバで利用を再開することにも当然に生じ得る。この場合、業務プロセスプログラムのそれぞれがアプリケーションプログラム（ユーザ認証プログラム、課金処理プログラム等）に組み込まれているので、アプリケーションサーバ単位で同一のユーザのユーザ認証がされ、同一のユーザに課金処理がされることになる。

【0045】従って、アプリケーション・サービス・プロバイダ(ASP)が同一のユーザに対して利用代金の支払いを求める場合、アプリケーション・サービス・プロバイダを行う事業者は同一のユーザに対する課金処理の結果をアプリケーションサーバのそれぞれで収集して利用代金の額を算出することになる。尚、課金処理の結果である利用代金の額は、アプリケーション・サービス・プロバイダを行う事業者とユーザとの契約でユーザが利用したアプリケーションプログラムの実績（利用時間、利用回数等）に基づいて算出されている。

【0046】ここで、アプリケーション・サービス・プロバイダ(ASP)とユーザとの間の契約で利用代金の額が、ユーザの利用時間（実績）に線形である場合、アプリケーションサーバ単位の課金処理の結果である利用代金の額を単に足すだけでよい。一例を挙げれば、契約で利用代金の額が、1000円/1時間当たり（ユーザの利用時間）であるとする。A（ユーザ）さんがA（アプリケーションプログラム）をAサーバ（アプリケーションサーバ）で1時間利用し、しかる後にAさんがAをBサーバで2時間利用したとする。Aサーバは課金処理の結果として利用代金の額を1000円であると算出し、Bサーバは課金処理の結果として利用代金の額を2000円であると算出する。これに対し、課金処理の結果である利用代金の額のそれぞれを単に足すだけでよい（1000円+2000円）。

【0047】しかしながら、アプリケーション・サービス・プロバイダ(ASP)とユーザとの間の契約で利用代金の額が、ユーザの利用時間（実績）に線形でない場合、アプリケーションサーバ単位の課金処理の結果である利用代金の額を単に足すことはできない。一例を挙げれば、契約で利用代金の額が、合計利用時間が1時間までは1000円/1時間当たりであり、合計利用時間が1時間以降2時間までは2000円/1時間当たりであるとする。先程と同様に、AさんがAをAサーバで1時間利用し、しかる後にAさんがAをBサーバで2時間利用したとする。Aサーバは課金処理の結果として利用代金の額を1000円であると算出し、Bサーバは課金処理の結果として利用代金の額を3000円（1000円+2000円）であると算出する。これに対し、課金処理の結果である利用代金の額のそれぞれを単に足すと4000円（1000円+3000円）。一方、契約ではAさんがAを3時間利用すると、契約の利用代金の額は5000円（1000×1+2000×2）である。

【0048】即ち、契約の利用代金の額がユーザの利用時間（実績）に線形でない場合、アプリケーションサーバ単位の課金処理の結果である利用代金の額を単に足すことはできない。従って、AさんはAをAサーバで1時間利用し、しかる後にAさんはAを再びAサーバで2時間利用しなければならない。即ち、同一のユーザが同一のアプリケーションプログラムを同一のアプリケーション

ンサーバで利用しなければならない。結果として、アプリケーションプログラムを複数のアプリケーションサーバに分散させても、一台のアプリケーションサーバに格納されるアプリケーションプログラムを利用するユーザの数を分散できない場合がある。これにより、アプリケーションプログラムの処理速度が低下することになる。

【0040】また請求項2の発明は、複数のアプリケーションサーバに格納されるアプリケーションプログラムと、それぞれのアプリケーションサーバに格納される一のアプリケーションサーバに送受信される第1のステップと、前記予め認証された認証証明書とサーバに予め格納される認証証明書と前記一のアプリケーションサーバによって比較されて一致する場合に、前記会話要求に対する会話返答と利用実績とが、前記一のアプリケーションサーバによって作成され、且つ前記会話返答と前記一のアプリケーションサーバによって再度認証された認証証明書と、前記一のアプリケーションサーバから前記ユーザ端末に送受信される第2のステップと、前記利用実績が、前記一のアプリケーションサーバから前記サーバに送受信される第3のステップと、を含むことを特徴とするアプリケーション・サービスを提供する方法である。

【0050】したがって請求項1の発明によれば、複数のアプリケーションサーバのそれぞれが、受信した認証証明書とサーバに予め格納される認証証明書とを比較して、受信した認証証明書の正当性の確認を行うことができる。即ち、複数のアプリケーションサーバのそれぞれが、同一のユーザを識別することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。これにより、アプリケーションプログラムを複数のアプリケーションサーバに分散させて、アプリケーションプログラムの処理速度の低下を防止することができるという利点がある。

【0051】また、請求項2の発明は、前記第3のステップ後に前記複数のアプリケーションサーバに格納されるアプリケーションプログラムのそれぞれのうち他のアプリケーションプログラムに対する会話要求と前記再度認証された認証証明書とが、前記ユーザ端末から前記一のアプリケーションサーバに格納される他のアプリケーションサーバに送受信される第4のステップと、前記再度認証された認証証明書と前記予め格納された認証証明書と前記他のアプリケーションサーバによって比較されて一致する場合に、前記会話要求に対する会話返答と利用実績とが、前記他のアプリケーションサーバによって作成され、且つ前記会話返答と前記他のアプリケーションサーバによって再度認証された認証証明書とが、前記他のアプリケーションサーバから前記ユーザ端末に送受信される第5のステップと、前記利用実績と

が、前記他のアプリケーションサーバから前記サーバに送受信される第6のステップと、を含むことを特徴とする第1項に記載のアプリケーション・サービスを提供する方法である。

【0052】したがって請求項2の発明によれば、第1項の発明の利点があるとともに、ユーザは一のアプリケーションプログラムによって再度認証された認証証明書を他のアプリケーションプログラムに送信することにより、他のアプリケーションプログラムを利用することができる。結果として、ユーザは同一のアプリケーションプログラムを他のアプリケーションサーバで利用することもでき、また、異なるアプリケーションプログラムを契約の範囲内で自由に利用することができる。このように、ユーザが同一のアプリケーションサーバだけに制限されることがないで、ユーザが複数のアプリケーションサーバに分散することになる。これにより、アプリケーションプログラムの処理速度の低下を防止することができるという利点がある。

【0053】また、請求項23の発明は、前記第3のステップ又は前記第6のステップ後に前記一のアプリケーションプログラムに対する会話要求と前記再度認証された認証証明書とが、ユーザ端末から前記一のアプリケーションサーバに送受信される第7のステップと、前記第7のステップ後に前記第2のステップ及び前記第3のステップが、1以上繰り返して実行される第8のステップと、を含むことを特徴とする第1項又は請求項22に記載のアプリケーション・サービスを提供する方法である。

【0054】したがって請求項23の発明によれば、第1項又は請求項22の発明の利点があるとともに、ユーザは一のアプリケーションサーバから他のアプリケーションサーバに接続した後、再び一のアプリケーションサーバに格納される同一のアプリケーションプログラムを利用することができる。例えば、ユーザが他のアプリケーションサーバに接続を切り換えたとしても、必ずしも他のアプリケーションサーバに格納される他のアプリケーションプログラムの処理速度が速いとは限らない。従って、他のアプリケーションプログラムの処理速度が遅い場合には、ユーザが一のアプリケーションプログラムの利用を再開することにより、他のアプリケーションプログラムの処理速度の更なる低下を防止することができる。

【0055】また、請求項24の発明は、前記一のアプリケーションサーバ及び/又は前記他のアプリケーションサーバから受信した利用実績に基づく利用代金の額が、前記サーバによって作成されるステップを含むことを特徴とする第1項から請求項23の何れか一に記載のアプリケーション・サービスを提供する方法である。

【0056】したがって請求項24の発明によれば、第1項から請求項23の何れか一の発明の利点があると

もに、利用実績がアプリケーションサーバのそれぞれからサーバに送信されるので、契約の利用代金の額がユーザの利用時間（実績）に依存しない場合であっても、サーバで利用実績を収集して、課金処理を行って利用代金の額を算出することができる。即ち、複数のアプリケーションサーバ間で課金管理の統一性を実現することができるという利点がある。

【0057】また、請求項25の発明は、アプリケーションプログラムの応答時間が、前記複数のアプリケーションサーバのそれぞれから前記サーバに定期的に送受信される監視ステップと、前記第1のステップ前にアプリケーションサーバに格納される接続要求とユーザ識別とが、前記ユーザ端末から前記複数のアプリケーションサーバのうち何れかの一のアプリケーションサーバに送受信されるログインステップと、前記ユーザ識別と前記サーバに予め格納されるユーザ識別とが前記何れか一のアプリケーションサーバによって比較されて一致する場合に、前記第1のステップに係る予め認証された認証証明書が、前記サーバに格納され、前記応答時間のそれぞれから前記サーバから前記何れか一のアプリケーションサーバに送受信されて、前記応答時間のそれぞれうち最も応答時間が短いアプリケーションプログラムが格納される最速アプリケーションサーバを特定し得るメニューが、作成され、且つ前記予め認証された認証証明書と前記メニューとが、前記何れか一のアプリケーションサーバから前記ユーザ端末に送受信される認証ステップと、を含み、前記メニューに基づいて前記最速アプリケーションサーバが前記ユーザ端末から接続されるとともに、前記第1のステップに係る一のアプリケーションサーバが前記最速アプリケーションサーバとして前記第1のステップが実行されることを特徴とする第1項から請求項24の何れか一に記載のアプリケーション・サービスを提供する方法である。

【0058】したがって請求項5の発明によれば、第1項から請求項24の何れか一の発明の利点があるとともに、アプリケーションプログラムの応答時間が監視されているので、アプリケーションプログラムの処理速度の低下を確実に防止することができるという利点がある。

【0059】また、請求項26の発明は、前記第2のステップ及び/又は前記第5のステップは、前記認証証明書と前記予め格納された認証証明書とが比較されて一致する場合に、前記予め格納される認証証明書が、新たな認証証明書に前記一のアプリケーションサーバ及び/又は前記他のアプリケーションサーバによって変更され、且つ前記新たな認証証明書が、前記サーバに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな認証証明書として送受信されることを特徴とする第1項から請求項25の何れか一に記載のアプリケーション・サービスを提供する方法である。

【0060】また、請求項27の発明は、前記第1のステップに係る認証証明書が、予め暗号化された暗号化済認証証明書であり、前記第2のステップ及び/又は前記第5のステップは、前記認証証明書と前記予め格納された認証証明書とが比較されて一致する場合に、前記予め格納された認証証明書が、新たな暗号化済認証証明書に前記一のアプリケーションサーバ及び/又は前記他のアプリケーションサーバによって暗号化され、且つ前記新たな暗号化済認証証明書が、前記サーバに格納されるステップを含み、前記再度認証された認証証明書が、前記新たな暗号化済認証証明書として送受信されることを特徴とする第1項から請求項26の何れか一に記載のアプリケーション・サービスを提供する方法である。

【0061】したがって請求項25又は請求項26の発明によれば、認証証明書の偽造を防止することができるという利点がある。

【0062】また、請求項28の発明は、複数のアプリケーションサーバに格納されるアプリケーションプログラムのそれぞれの一のアプリケーションサーバが格納される一のアプリケーションサーバを選択し、前記一のアプリケーションプログラムに対する会話要求と予め認証された認証証明書とを前記一のアプリケーションサーバに送信し、且つ前記会話要求に対する会話返答と前記一のアプリケーションサーバによって再度認証された認証証明書とを前記一のアプリケーションサーバから受信する手段を備えるユーザ端末と、前記会話要求と前記予め認証された認証証明書とを前記ユーザ端末から受信し、前記予め認証された認証証明書とサーバに予め格納される認証証明書とを比較して一致する場合に、前記会話返答と前記会話要求に対する利用実績とを作成し、前記会話返答と再度認証された認証証明書とを前記ユーザ端末に送信し、且つ利用実績を前記サーバに送信する手段をそれぞれ備える複数のアプリケーションサーバと、前記認証された認証証明書を予め格納し、且つ前記利用実績を前記一のアプリケーションサーバから受信する手段を備えるサーバと、から構成されてなるアプリケーション・サービスを提供するシステムである。

【0063】したがって請求項28の発明によれば、複数のアプリケーションサーバのそれぞれが、受信した認証証明書とサーバに予め格納される認証証明書とを比較して、受信した認証証明書の正当性の確認を行うことができる。即ち、複数のアプリケーションサーバのそれぞれが、同一のユーザを識別することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。これにより、アプリケーションプログラムを複数のアプリケーションサーバに分散させて、アプリケーションプログラムの処理速度の低下を防止することができるという利点がある。

【0064】また、請求項29の発明は、前記サーバが、前記利用実績に基づく利用代金の額を作成する手段

を備えることを特徴とする請求項 2 に記載のアプリケーション・サービスを提供するシステムである。

【0065】したがって請求項 2 の発明によれば、請求項 2 の発明の利点があるとともに、利用実績がアプリケーションサーバのそれぞれからサーバに送信されるので、契約の利用代金の額がユーザの利用時間（実績）に振形でない場合であっても、サーバで利用実績を収集して、課金処理を行って利用代金の額を算出することができる。即ち、複数のアプリケーションサーバ間で課金管理の統一性を実現することができるという利点がある。

【0066】また、請求項 3 の発明は、ユーザ端末が、アプリケーションプログラムに対する接続要求とユーザ識別とを前記複数のアプリケーションサーバのうち何れか一アプリケーションサーバに送信し、且つ前記予め認証された認証証明書と前記応答時間のそれぞれのうち最も応答時間が短いアプリケーションプログラムが格納される最速アプリケーションサーバを特定し得るメニューとを前記何れか一のアプリケーションサーバから受信して、前記一のアプリケーションサーバを前記最速アプリケーションサーバとして選択する手段を備え、前記複数のアプリケーションサーバのそれぞれが、アプリケーションプログラムの応答時間を前記サーバに定期的に送信し、前記接続要求と前記ユーザ識別とを前記ユーザ端末から受信して、前記ユーザ識別とサーバに予め格納されるユーザ識別とを比較して一致する場合に、前記予め認証された認証証明書を前記サーバに格納し、前記応答時間のそれぞれを前記サーバから受信して、前記メニューを作成し、且つ前記予め認証された認証証明書と前記メニューとを前記ユーザ端末に送信する手段を備え、前記サーバが、前記応答時間を前記複数のアプリケーションサーバのそれぞれから定期的に受信し、前記ユーザ識別を予め格納し、且つ前記応答時間のそれぞれを前記アプリケーションサーバのそれぞれに送信する手段を備えることを特徴とする請求項 2 又は請求項 2 に記載のアプリケーション・サービスを提供するシステムである。

【0067】したがって請求項 3 の発明によれば、請求項 2 又は請求項 2 の発明の利点があるとともに、アプリケーションプログラムの応答時間が監視されているので、アプリケーションプログラムの処理速度の低下を確実に防止することができるという利点がある。

【0068】また、請求項 3 の発明は、前記複数のアプリケーションサーバのそれぞれが、前記認証証明書と前記予め格納された認証証明書とを比較して一致の場合に、前記予め格納された認証証明書と新たな認証証明書とを交換して、前記新たな認証証明書を格納する手段を備え、前記再度認証された認証証明書を前記新たな認証証明書として送信することを特徴とする請求項 2 から請求項 3 の何れか一に記載のアプリケーション・サー

ビスを提供するシステムである。

【0069】また、請求項 3 の発明は、前記ユーザ端末が送信する前記予め認証された認証証明書が、予め暗号化された暗号化済認証証明書であり、前記複数のアプリケーションサーバのそれぞれが、前記認証証明書と予め格納された認証証明書とを比較して一致する場合に、前記予め格納された認証証明書を新たな暗号化済認証証明書に暗号化して、前記新たな暗号化済認証証明書を格納する手段を備え、前記再度認証された認証証明書を前記新たな暗号化済認証証明書として送信することを特徴とする請求項 2 から請求項 3 の何れか一に記載のアプリケーション・サービスを提供するシステムである。

【0070】したがって請求項 3 又は請求項 3 の発明によれば、認証証明書の偽造を防止することができるという利点がある。

【発明の実施の形態】以下に本発明の実施の形態のユーザの認証及び課金方法、その記録媒体、アプリケーション・サービスを提供する方法及びシステムにつき図面を参照して説明する。

【0071】（実施の形態 1）まず、本発明の実施の形態 1 のアプリケーション・サービスを提供するシステムにつき、図 5 を参照して、説明する。図 5 は本発明の実施の形態 1 のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

【0072】まず、図 5 を参照すると、本実施の形態 1 のアプリケーション・サービスを提供するシステムは、アプリケーションプログラムを利用するユーザ端末 100 と、アプリケーションプログラムが動作するアプリケーションサーバ 1 (300) と、各種のを格納するサーバ 300 と、から構成されてなる。ここで、ユーザ端末 100 と、アプリケーションサーバ 1 (300) とは、コンピュータネットワーク 200 を介して相互に接続可能に構成されている。また、サーバ 500 と、アプリケーションサーバ 1 (300) とは、主として同一組織内で用いられる総合的な情報通信ネットワーク (LAN: Local Area Network)、公衆回線、又は専用線を介して相互に接続可能に構成されている。尚、ユーザ端末 100 と、アプリケーションサーバ 1 (300) と、サーバ 500 とが、コンピュータネットワーク 200 を介して相互に接続可能に構成されていてもよい。また、コンピュータネットワーク 200 とは、例えば、インターネット (Internet)、広域通信網 (WAN: Wide Area Network) 等である。更に、コンピュータネットワーク 200、LAN 等は、有線、無線を問わず、ユーザ端末 100 とアプリケーションサーバ 1 (300) との間、ユーザ端末 100 とサーバ 500 との間、アプリケーションサーバ 1 (300) とサーバ 500 との間、及びユーザ端末 100 とアプリケーションサーバ 1 (300) とサーバ 500 との間のそれぞれで送受信するためのものであればよい。

【0073】ユーザ端末 100 は、コンピュータネット

ワーク(200)を介してホームページへのアクセス等を可能とするWebブラウザ(101)と、ホームページを表示可能とする表示手段(図示せず)、文字・記号等を入力するキーボード(ボタン)、マウス等の入力手段(図示せず)とを備える。ここで、ユーザ端末(100)は、例えば、パソコン(デスクトップ型パソコン、ノート型パソコン)、携帯電話機等である。

【0074】アプリケーションサーバ1(300)は、HTML、文書等のリソース(図示せず)を格納する記憶手段と、ユーザを認証する認証システム(301)と、ユーザ個別のメニューを動的に生成する動的メニュー生成システム(302)と、コンピュータネットワーク(200)を介して有償で提供する複数のアプリケーションプログラムA(310)及びアプリケーションプログラムB(320)と、Webサーバ(390)とを備える。また、アプリケーションプログラムA(310)及びアプリケーションプログラムB(320)のそれぞれは、認証API(303)と課金API(304)とを備える。更に、Webサーバ(390)は、ユーザ端末(100)のWebブラウザ(101)とHTTTPで通信する。即ち、Webサーバ(390)は、認証システム(301)、動的メニュー生成システム(302)、アプリケーションプログラムA(310)、及びアプリケーションプログラムB(320)のそれぞれと、Webブラウザ(101)との通信を中継する。ここで、アプリケーションサーバ1(300)は、例えば、いわゆるサーバ(ハードディスク等の記憶手段の容量が大きいオフコン)、パソコン等である。

【0075】サーバ(500)は、ユーザ認証に必要な(ユーザID、パスワード、端末識別子等を含む。)を格納する認証保持システム(501)と、ユーザごとにあらかじめ合意した利用可能なアプリケーションプログラムのリストを格納するメニュー保持システム(502)と、利用実績を蓄積してユーザ別アプリケーションプログラム別請求書を作成する課金システム(503)とを備える。ここで、サーバ(500)は、例えば、いわゆるサーバ、パソコン等である。

【0076】尚、本実施の形態1のアプリケーション・サービスを提供するシステムの構成要素のうち、認証システム(301)と、動的メニュー生成システム(302)と、認証API(303)と、課金API(304)と、認証保持システム(501)と、メニュー保持システム(502)と、課金システム(503)とをASP基盤システム(500)とする。また、ASP基盤システムを格納するサーバをASP基盤サーバ(900)という。

【0077】次に、本発明の実施の形態のアプリケーション・サービスを提供する方法及びシステムの前提となるアプリケーション・サービスを提供するための準備をするシステム及び方法につき、図1～図4を参照して、説明する。尚、アプリケーション・サービスを提供するための準備をするシステム方法は、アプリケーションプログラムを準備する方法(ステップA1～A7)と、ア

プリケーションプログラムを利用するユーザの登録方法(ステップB1～B5)と、から成る。図1はアプリケーション・サービスを提供するための準備をするシステムの構成例を示すブロック図である。図2はアプリケーションプログラムを準備する方法を説明するためのフローチャート図である。

【0078】予め、ASP基盤システムのうち認証システム(301)と、動的メニュー生成システム(302)と、認証API(303)と、課金API(304)とが、アプリケーションサーバ1(300)に格納されている。また、ASP基盤システムの残りの認証保持システム(501)と、メニュー保持システム(502)と、課金システム(503)とが、ASP基盤サーバ(900)に格納されている。加えて、開発されたアプリケーションプログラムが、ベンダー端末(700,701)に格納されている。

【0079】[ステップA1]

(アプリケーションプログラムを準備する方法)ASP基盤サーバ(900)に係るASP基盤システム(認証システム(301)、動的メニュー生成システム(302)、認証API(303)、課金API(304))が、公衆回線若しくは専用線(220)、又は光磁気ディスク(MO:Magneto Optical disk)、光ディスク(CD:Compact Disk, DVD:Digital Versatile Disk)、フロッピー(登録商標)ディスク(FD:Floppy Disk)等の補助記憶媒体(230)を介してASP基盤サーバ(900)からベンダー端末(700,710)に提供されて格納される。

【0080】[ステップA2]開発されたアプリケーションプログラムAをASP基盤システムに適合させるため、提供された認証API(303)及び課金API(304)が、ベンダー端末(700)によって開発されたアプリケーションプログラムA(310)に組み込まれる。

【0081】[ステップA3]アプリケーションサーバ1(300)でアプリケーションプログラムのサービスのを行うため、提供された認証システム(301)及び動的メニュー生成システム(302)が、LAN(210)を介してベンダー端末(700)からアプリケーションサーバ2(400)にインストールされて格納される。尚、ここで、ASP基盤システムの認証システム(301)及び動的メニュー生成システム(302)が、公衆回線等(220,230)を介してアプリケーションサーバ2(400)にインストールされてもよいことは言うまでもない。

【0082】[ステップA4]認証API(303)及び課金API(304)が組み込まれたアプリケーションプログラムA(310)が、ベンダー端末(700)からアプリケーションサーバ1(300)にLAN等(210,220,230)を介してインストールされて格納される。

【0083】[ステップA5]開発されたアプリケーションプログラムBをASP基盤システムに適合させるため、提供された認証API(303)及び課金API(304)が、ベンダー端末(701)によって開発されたアプリケーション

ジョンプログラムB(320)に組み込まれる。

【0084】ステップA6「認証API(303)及び課金API(304)が組み込まれたアプリケーションプログラムB(320)が、ペンダント端末(710)からアプリケーションサーバ1(300)にLAN等(210,220,230)を介してインストールされて格納される。

【0085】ステップA7「アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)及びアプリケーションプログラムB(320)のアプリケーション情報(機能概要、詳細説明、デモプログラムへのリンク、契約条件(料金表を含む))が、サーバ(500)に登録されて格納される。

【0086】尚、アプリケーションサーバ1(300)に複数のアプリケーションプログラム(310,320,...)が格納される場合には、アプリケーションプログラム(310,320,...)のそれぞれに対するアプリケーション情報がサーバ(500)に登録されて格納される。

【0087】また、図3はアプリケーションプログラムを利用するユーザの登録方法を説明するためのフローチャート図である。

【0088】[ステップB1]

(アプリケーションプログラムを利用するユーザの登録方法) 先ず、ユーザ端末(100)はサーバ(500)に接続する。

【0089】[ステップB2]これに対し、サーバ(500)はアプリケーション情報をユーザ端末(100)に送信する。尚、サーバ(500)は、ユーザ端末(100)から入力される検索条件とアプリケーション情報に含まれる機能概要の一部とが一致するアプリケーション情報を検索し、検索条件に一致するアプリケーション情報をユーザ端末(100)に送信する手段を備える。

【0090】[ステップB3]ユーザ端末(100)は検索条件を入力してアプリケーション情報を受信してユーザ端末(100)の表示手段(図示せず)に表示する。ここで、例えば、図4に示すアプリケーション情報表示画面が、ユーザ端末(100)の表示手段に表示される。図4を参照するに、表示されるアプリケーション情報表示画面は、アプリケーション名称(例えば、日英辞典、百科事典)、機能概要の一覧である。ここで、ユーザは表示手段に表示される「詳細表示ボタン」、「デモ表示ボタン」、及び「料金表示ボタン」のそれぞれのボタンをマウスでクリックすることができる。「詳細表示ボタン」がクリックされると、アプリケーションプログラムの機能詳細説明画面(図示せず)が表示手段に表示される。

また、「デモ表示ボタン」がクリックされると、アプリケーションプログラムのアプリケーションプログラムのデモプログラムが実行され、アプリケーションデモ画面(図示せず)が表示手段に表示される。加えて、「詳細表示ボタン」がクリックされると、料金表等の契約条件を表す契約条件画面(図示せず)が表示手段に表示され

る。

【0091】[ステップB4]ユーザは、利用するアプリケーションプログラム(百科事典)を決めたら、当該アプリケーションプログラムの「ラジオボタン(選択)」をチェックして(図4中の●)、「契約申し込みボタン」をクリックする。その後、ユーザ登録画面(図示せず)がユーザ端末(100)の表示手段に表示され、ユーザは登録に必要なユーザ情報(企業名、責任者名、住所、電話番号、決済方法、端末識別子等)を入力してサーバ(500)に送信する。ここで、端末識別子とは、ユーザ端末(100)のIPアドレス(IP address)若しくはフル・クォリファイド・ドメイン・ネーム(FQDN:Full Qualified Domain Name)である。例えば、IPアドレスは192.168.01.001であり、FQDNはwww.○○○.co.jpである。

【0092】[ステップB5]これに対し、サーバ(500)は必要なユーザ情報をユーザ端末(100)から受信し、利用に必要なユーザ情報(ユーザID(ユーザ識別)、パスワード)を発行する。サーバ(500)は、登録に必要なユーザ情報に前記利用に必要なユーザ情報を加えたユーザ情報(ユーザID、パスワード、企業名、責任者名、住所、電話番号、決済方法、端末識別子等)を登録して格納する。

【0093】[ステップB6]サーバ(500)は、利用に必要な情報(ユーザID、パスワード)をユーザ端末(100)に送信して通知する。

【0094】次に、[ステップA2]に係る認証API(303)及び課金API(304)をアプリケーションプログラムに組み込む方法につき、図6及び図7を参照して、説明する。図6は認証API(303)をアプリケーションプログラムに組み込む方法を示すフローチャートである。このフローチャートで認証API(303)をアプリケーションプログラムに組み込む方法を説明する。まず組み込む前の一般的なアプリケーションプログラムの入力メッセージを処理するフローを説明する。

【0095】[ステップG1]アプリケーションプログラムは、ユーザ端末(100)からメッセージ(会話要求)を受信する。

[ステップG2]アプリケーションプログラムは、受信したメッセージIDが処理1Dに等しいか比較する。受信したメッセージIDが処理1Dに等しければ該当する処理1を実行する(ステップG5へ)。受信したメッセージIDが処理1Dに等しくなければ次の処理1Dに等しいか比較する(ステップG3へ)。

[ステップG3]アプリケーションプログラムは、受信したメッセージIDが処理2Dに等しいか比較する。受信したメッセージIDが処理2Dに等しければ該当する処理2を実行する(ステップG6へ)。受信したメッセージIDが処理2Dに等しくなければ次の処理1Dに等しいか比較する(ステップG4へ)。

[ステップG4]アプリケーションプログラムは、受信し

たメッセージ1Dが処理nに等しいか比較する。受信したメッセージ1Dが処理nに等しければ該当する処理nを実行する(ステップG7へ)。受信したメッセージ1Dが処理nに等しくなければエラー処理を行う(ステップG8へ)。

[ステップG5]アプリケーションプログラムは、受信したメッセージを処理1して結果をHTML文書形式にしてユーザ端末100に送信する。

[ステップG6]アプリケーションプログラムは、受信したメッセージを処理2して結果をHTML文書形式にしてユーザ端末100に送信する。

[ステップG7]アプリケーションプログラムは、受信したメッセージを処理nして結果をHTML文書形式にしてユーザ端末100に送信する。

[ステップG8]アプリケーションプログラムは、受信したメッセージ1Dが最後の処理1Dと一致しなければエラーメッセージをユーザ端末100に送信する。

【0096】次に認証API(303)をアプリケーションプログラムに組み込む方法を説明する。ステップG1とステップG2の間に次に示すステップH1～H8を挿入する。尚、ステップC5において、動的メニュー生成システム302が暗号化された認証証明書ユーザ端末100に送信され、ステップC8において、暗号化された認証証明書がユーザ端末100からアプリケーションプログラムに送信されているとする。

[ステップH1]アプリケーションプログラムは、暗号化された認証証明書をクッキーとしてユーザ端末100から受信する。

[ステップH2]アプリケーションプログラムは、暗号化された認証証明書を認証API(303)の引数にセットする。

[ステップH3]アプリケーションプログラムは、認証API(303)を呼び出す。

[ステップH4]アプリケーションプログラムは、認証API(303)から戻された値をチェックする。無効コードならステップH5へ行く。

[ステップH5]アプリケーションプログラムは、エラーメッセージとログイン画面を表示する。

[ステップH6]アプリケーションプログラムは、認証API(303)から戻された新しい暗号化された認証証明書をクッキーとしてユーザ端末100に送信する。尚、ステップH1～ステップH8を実行させるためのプログラムを認証関数プログラムという。

【0097】図7は課金API(304)をアプリケーションプログラムに組み込む方法を示すフローチャートである。このフローチャートで認証API(303)をアプリケーションプログラムに組み込む方法を説明する。ステップ12が本平のアプリケーションの処理であり、図6のステップH5～H7が示す処理である。ステップ12(ステップH5～H7)の直前にステップH1を挿入

し、ステップ12の後ろにステップ13～ステップ15を挿入する。

[ステップ11]アプリケーションプログラムは、ステップ12の直前に処理開始時刻としてマシン時間を取得する。

[ステップ12]アプリケーションプログラムは、処理をするとともに、課金対象となる使用量を計測する。

[ステップ13]アプリケーションプログラムは、ステップ12の直後に処理終了時刻としてマシン時間を取得する。

[ステップ14]アプリケーションプログラムは、利用実績(ユーザ識別名、処理開始時刻、処理時間(=処理終了時刻-処理開始時刻)、ステップ12で採取した課金対象となる使用量等を含む。)を作成する。(図10に示す利用実績(503-2)を参照。)

[ステップ15]アプリケーションプログラムは、課金API(304)を呼び出す。尚、ステップ11～ステップ15を実行させるためのプログラムを課金関数プログラムという。

【0098】次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図5、図8～図10を参照して、説明する。

【0099】[ステップL1]ユーザは、ユーザ端末100のWebブラウザ101を使用してコンピュータネットワーク200上に開設しているアプリケーションサーバ130に接続する。

【0100】[ステップL2]認証システム301が、Webサーバ390を介してログイン画面をユーザ端末100に送出する。

【0101】[ステップL3]ユーザは、ユーザ端末100の表示手段に表示されたログイン画面からログインID(ユーザ識別)とパスワードとを入力する。

【0102】[ステップL4]認証システム301は、ユーザ端末100から受信したログインIDとパスワードとユーザ端末100の端末識別子とをWebサーバ390から受け取る。ユーザ識別名、パスワード、端末識別子のすべてまたはあらかじめ決めてあるそれらの内のいくつかの組み合わせと、認証保持システム501からログインIDをキーにして受信した認証(ユーザID、パスワード、端末識別子を含む。図8、認証501-1参照。)を突き合わせて一致するか比較する。一致すれば正常、一致しなければ以上と判断する。

【0103】[ステップL5]認証システム301の認証結果が正常なら、認証したときの時刻(年月日時分秒)をマシン時間から取得する。次に、認証システム301は、ユーザIDと認証したときの時刻を含む認証証明書を作成する。同時に、認証システム301は、認証証明書を暗号化し、暗号化した認証証明書を作成する。次に、認証システム301は、暗号化した認証証明書をキーとしてユーザ認証保持システム501に暗号化前の認

証明書と暗号化した認証証明書と(図8. 認証証明書501参照。)を保存する。そして認証システム(301)は動的メニュー生成システム(302)を起動し、ユーザIDと暗号化した認証証明書を引き出す。ここで、認証証明書の暗号化には、第三者に解読不可能な方向暗号アルゴリズムを使用する。たとえば変換関数MD5(RC1321で定義されている)を使う。尚、MD5の出力結果はバイナリなのでネットワーク透過性のためにBASE64でエンコードするものを暗号化した証明書とする。

【0104】[ステップL6]動的メニュー生成システム(302)は、ユーザIDをキーとしてメニュー保持システム(502)から当該ユーザが利用可能なアプリケーションプログラムのリスト(図9. アプリケーションプログラムリスト502-1参照。)と各アプリケーションプログラムが動作可能なサーバのリスト(図9. サーバリスト502-2参照。)を受信する。アプリケーションプログラムリスト(502-1)は、ユーザIDとアプリケーションプログラム名を含む。また、サーバリスト(502-2)は、アプリケーションプログラム名とサーバ端末識別名とプログラムファイル名を含む。

【0105】[ステップL7]動的メニュー生成システム(302)はクッキー化した、且つ暗号化した認証証明書と当該ユーザが利用可能なアプリケーションプログラムのリストをURI形式に合成し、メニューを生成する。動的メニュー生成システム(302)は生成したメニューをWebサーバ(390)を介してユーザ端末(100)に送出する。

【0106】[ステップL8]ユーザが、メニュー画面からアプリケーションA(310)を選択すると、Webブラウザ(101)はアプリケーションA(310)のURIを使って選択されたアプリケーションに接続する。このとき、ユーザ端末(100)は、ステップL7で受信した暗号化された認証証明書をクッキーとして送出する。

【0107】[ステップL9]アプリケーションA(310)は、Webサーバ(390)を介してクッキーとして暗号化された認証証明書を受信する。アプリケーションA(310)は、この暗号化された認証証明書を引数に格納して認証API(303)を呼び出す。認証API(303)は、認証システム(310)に暗号化された認証証明書を送出し、正当性確認を要求する。

【0108】[ステップL10]認証システム(301)は、暗号化された認証証明書をキーとして認証保持システム(501)から認証証明書を取り出す。認証証明書に含まれる認証時刻が、現在のマシン時刻を基点としてあらかじめ決めておいた無通信タイムアウト時間より過去であれば、この暗号化された認証証明書は無効であると判定する。認証システム(301)は、保持システム(501)から取り出した認証証明書を再度暗号化したものが受信した暗号化された認証証明書と一致すれば有効、一致しなければ無効と判定する。認証システム(301)の認証結果が有効であれば、認証システム(301)は、認証証明書の認証時

刻を現在のマシン時刻に変更して認証証明書を更新する。次に、認証システム(301)は、新しい認証証明書を暗号化する。認証システム(301)は、この暗号化した認証証明書をキーとして新しい認証証明を認証保持システム(401)に登録し、古い認証証明は削除する。

【0109】[ステップL11]認証システム(301)は、認証API(303)を介してアプリケーションA(310)に認証結果を応答する。認証システム(301)の認証結果が有効なら、ユーザIDと新しい暗号化された認証証明書を返し、無効なら無効を意味するコードを返す。

【0110】[ステップL12]アプリケーションA(310)は、認証システム(301)の認証結果が正常なら、ユーザ端末(100)とアプリケーションレベルの会話(メッセージ)を開始する。このとき、アプリケーションA(310)は、ユーザ端末(100)からWebサーバ(390)を介してを受信すると、必ず暗号化された認証証明書がクッキーとして送られてくるので、毎回ステップL9～ステップL11の手順で正当性を確認する。アプリケーションA(310)は、ユーザ端末(100)にWebサーバ(390)を介してHTML文書を送信するときはステップL11で返された新しい暗号化された認証証明書をクッキーとして送信する。

【0111】[ステップL13]一連のアプリケーションレベルの会話(メッセージ: 会話要求、会話返送)の途中、又はアプリケーションA(310)の終了時に、アプリケーションA(310)は利用実績(503-2)としてユーザ識別名、時刻、利用機能、各アプリケーションプログラムで決めた課金対象となるアプリケーション利用量等を課金API(304)を介して課金システム(503)に送出する。課金システム(503)は、受信した利用実績を蓄積し、定期的に集計して、認証保持システム(501)に格納しているユーザ別アプリケーション別各アプリケーション利用量単価(図10. 単価マスタ503-1参照。)にユーザ別アプリケーション別アプリケーション利用量を横断し、利用料金の額を算出して請求書を作成する。

【0112】(実施の形態2)次に、本発明の実施の形態2のアプリケーション・サービスを提供する方法及びシステムにつき、図11を参照して、説明する。図11は本発明の実施の形態2のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。図5に示す本発明の実施の形態1のシステムとはアプリケーションサーバ1(300)から構成されていたが、図12に示す実施の形態2のシステムは複数のアプリケーションサーバ(300, 400, ...)から構成されている。

【0113】次に、本発明の実施の形態2のアプリケーション・サービスを提供する方法につき、図11を参照して、説明する。尚、ステップL1～ステップL13に相当するものである。

【0114】[ステップC1: ログインステップ]アプリケーションプログラムに対する接続要求が、ユーザ端末

(100)から複数のアプリケーションサーバ(300,400,...)のうち何れかのアプリケーションサーバ1(300)に送受信される。ここで、アプリケーションサーバ1(300)は、接続要求をWebサーバ(390)を介して認証システム(301)に受信している。これに対し、アプリケーションサーバ1(300)は、ログイン画面をWebサーバ(390)を介して認証システム(301)からユーザ端末(100)に送信する。一方、ユーザ端末(100)は、Webブラウザ(101)を介してログイン画面を受信すると、ログイン画面がユーザ端末(100)の表示手段に表示される。次に、ユーザは、ログインID(ユーザ識別)とパスワードとを入力してアプリケーションサーバ1(300)に送信する。これに対し、アプリケーションサーバ1(300)は、ログインID(ユーザ識別)とパスワードとをWebサーバ(390)を介して認証システム(301)に入力する。

【0115】尚、ログインID(ユーザ識別)とパスワードとのバケットがユーザ端末(100)からアプリケーションサーバ1(300)にIPプロトコルで送受信されているので、バケットは常に送信元(ユーザ端末(100)のIPアドレス)と送信先(アプリケーションサーバ1(300)のIPプロトコル)とが格納されている。従って、アプリケーションサーバ1(300)は、ログインID(ユーザ識別)とパスワードとを受信するとともに、ユーザ端末(100)のIPアドレス(端末識別子)をバケットから抽出して認証システム(301)に入力することができる。

【0116】ステップC2(認証ステップ)次に、承認システム(301)は、入力されたログインID(ユーザ識別)及びパスワードと、認証保持システム(501)に予め格納されるユーザID(ユーザ識別)及びパスワードと、と比較する。この比較の結果として、入力されたログインID及びパスワードと、予め格納されるユーザID及びパスワードと、が一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを認証する。この時、認証システム(301)は、ユーザIDと認証した時刻とを含む認証証明書を作成して認証保持システム(501)に格納する。ここで、認証証明書が認証した時刻を含むことにより、認証証明書に有効期限を持たせることができる。また、認証証明書がユーザIDとともに認証した時刻を含むことにより、承認時に毎回異なるユニークな認証証明書を作成することができる。認証証明書が有効期限で時間で管理される、又は認証証明書が承認時に毎回異なるので、認証証明書の偽造を防止することができるという利点がある。加えて、認証システム(301)が暗号化した認証証明書を作成することにより、認証証明書の偽造を防止を強化することができる。

【0117】尚、認証システム(301)は、IPアドレス(端末識別子)が入力される場合には、入力されたIPアドレス(端末識別子)と、予め格納される端末識別子と、を更に比較することができる。この比較の結果とし

て、入力されたログインID、パスワード、及び端末識別子と、予め格納されるユーザID、パスワード、及び端末識別子と、が完全に一致する場合に、認証システム(301)は、ユーザ端末(100)のユーザを認証することもできる。これにより、認証システム(301)は、仮に入力されたログインID及びパスワードが偽造されたものであっても、正規なユーザのみを正確に認証することができる。

【0118】ステップC3次に、認証システム(301)は、動的メニュー生成システム(302)を起動し、認証証明書と動的メニュー生成システム(302)に入力する。【0119】ステップC4これに対し、動的メニュー生成システム(302)は、入力された認証証明書のユーザIDをキーとして、予め当該ユーザが利用契約したアプリケーションプログラムリストをメニュー保持システム(702)から取得する。次に、動的メニュー生成システム(302)は、取得したアプリケーションプログラムリストのアプリケーションプログラム名をキーとしてサーバリストをメニュー保持システム(702)から取得する。ここで、取得したサーバリストが複数ある場合、動的メニュー生成システム(302)は、エラーステータス及び平均応答時間を複数のサーバリストから取得する。次に、動的メニュー生成システム(302)は、抽出したエラーステータスがサービス中であり、且つ抽出した平均応答時間が最小であるサーバリストを選択する。次に、動的メニュー生成システム(302)は、選択したサーバリストに係るプログラムファイル名及びサーバ端末識別名を取得する。次に、動的メニュー生成システム(302)は、取得したプログラムファイル名及びサーバ端末識別名に基づいてURI(Universal Resource Identifier)情報を含むメニューを動的に作成する。尚、URIは、IETFが公開している技術文書rfc2396で定義されている(<http://www.ietf.org/rfc/rfc2396.txt>)。また、URIはURI(Universal Resource Locator)と同義である。

【0120】ステップC5次に、動的メニュー生成システム(302)は、メニューと認証証明書とをWebサーバ(390)を介してユーザ端末(100)に送信する。尚、認証証明書が暗号化されている場合には、暗号化された認証証明書のみがユーザ端末(100)に送信される。これに対し、ユーザ端末(100)は、メニューと認証証明書とをWebブラウザ(101)を介してWebサーバ(390)から受信する。ここで、Webブラウザ(101)は、認証証明書をクッキー(cookie)として一次保持する。

【0121】ステップC6ユーザが、ユーザ端末(100)でアプリケーションプログラムA(310)を選択すると、メニューのURIに従って、ユーザ端末(100)は、アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)に対する会話要求(メッセージ)をWebブラウザ(101)を介してアプリケーションプログラムA(310)に送信する。この時、ユーザ端末(10

0)は、認証証明書をクリック(Cookie)として送信する。これに対し、アプリケーションプログラムA(310)は、会話要求(メッセージ)と認証証明書とをWebサーバ(390)を介してアプリケーションプログラムA(310)で受信する。

【0122】[ステップC7]アプリケーションプログラムA(310)は、認証証明書をアプリケーションプログラムA(310)に組み込まれた認証API(303)にセットする。ここで、API(Applcation Program Interface)とは、認証システム(301)、課金システムプログラム(503)等の外部プログラムをアプリケーションプログラムA(310)内部から使用するための仕組みである。尚、具体的に言えば、APIはクラス名、メソッド名、引数の並びで規定されている。即ち、アプリケーションプログラムA(310)は、認証証明書を認証API(303)を介して認証システム(301)に入力する。ここで、認証API(303)は、認証証明書を認証システム(301)に送り、認証証明書の正当性を確認している。

【0123】[ステップC8:再認証ステップ]これに対し、認証システム(301)は、認証API(303)を介して入力された認証証明書と、予め認証した認証証明書と、と比較する。この比較の結果として、入力された認証証明書と、予め格納した認証証明書と、が一致する場合には、認証システム(301)は、ユーザ端末(100)のユーザを再度認証する。この時、認証時刻(301)は、ユーザIDと再度認証した時刻を含む認証証明書を作成して認証保持システム(501)に格納する。即ち、認証システム(301)は、認証証明書の認証時刻を更新して、認証保持システム(501)に格納する。次に、認証システム(301)は、再度認証された認証証明書を認証API(303)を介してアプリケーションプログラムA(310)に出力する。

【0124】[ステップC9]次に、アプリケーションプログラムA(310)は、会話要求(メッセージ)に対する会話返答(メッセージ)を作成する。次に、アプリケーションプログラムA(310)は、作成した会話返答(メッセージ)と再度認証された認証証明書とをWebサーバ(390)を介してユーザ端末(100)に送信する。これに対し、ユーザ端末(100)は、作成した会話返答(メッセージ)と再度認証された認証証明書とをWebブラウザ(101)を介してWebサーバ(390)から受信する。ここで、再度認証された認証証明書は、Webサーバ(390)からWebブラウザ(101)にCookieとして送受信されている。この後、ユーザ端末(100)とアプリケーションA(310)の間でメッセージの送受信があるたびに、認証証明書の認証及び更新を行う。これらにより、認証証明書の偽造を防止を更に強化することができる。

【0125】[ステップC10]ユーザがアプリケーションプログラムA(310)の利用を終了すると、利用実績を課金API(304)を介して課金システム(503)に出力する。

【0126】[ステップC11]前記ステップC6のメニューに含まれるURIは、アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)を特定し得るものであった。即ち、ステップC4において、アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)の平均応答時間が最小であったので、URIは、アプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)を特定し得るものであった。ところが、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)の平均応答時間が最小である場合、動的メニュー生成システム(302)が作成するメニューに係るURIは、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)を特定し得るものになる。この場合、ユーザが、ユーザ端末(100)でアプリケーションプログラムA(310)を選択すると、メニューのURIに従って、ユーザ端末(100)は、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)に対する会話要求(メッセージ)をWebブラウザ(101)を介してアプリケーションプログラムA(310)に送信する。この時、ユーザ端末(100)は、認証証明書をクリック(Cookie)として送信する。これに対し、アプリケーションサーバ2(400)に格納されるアプリケーションプログラムA(310)は、会話要求(メッセージ)と認証証明書とをWebサーバ(390)を介してアプリケーションプログラムA(310)で受信する。

【0127】[ステップC12]アプリケーションプログラムA(310)は、認証証明書を認証API(303)を介して認証システム(301)に入力する。

【0128】[ステップC13]これに対し、認証システム(301)は、認証API(303)を介して入力された認証証明書と、予め認証した認証証明書と、と比較する。この比較の結果として、入力された認証証明書と、予め格納した認証証明書と、が一致する場合には、認証システム(301)は、ユーザ端末(100)のユーザを再度認証する。この時、認証システム(301)は、ユーザIDと再度認証した時刻を含む認証証明書を作成して認証保持システム(501)に格納する。次に、認証システム(301)は、再度認証された認証証明書を認証API(303)を介してアプリケーションプログラムA(310)に出力する。

【0129】このように、アプリケーションサーバ1(300)及びアプリケーションサーバ2(400)は、認証証明書をCookieの仕組みで受信し、認証システム(301)を介して認証保持システム(501)に格納される認証証明書と比較して受信した認証証明書の正当性の確認を行うことができる。即ち、最初にユーザがログインしたアプリケーションサーバ1(300)に格納されるアプリケーションプログラムA(310)のほか、最初にログインしたアプリケーションサーバ1(300)と異なるアプリケーションサーバ2(400)に格納されるアプリケーションプログラム

A (310)も同一のユーザを識別することができる。換言すれば、複数のアプリケーションサーバ間でユーザ管理の統一性を実現することができる。

【0130】ステップ14]ユーザがアプリケーションプログラムA (310)の利用を終了すると、利用実績を課金API (304)を介して課金システム503に出力する。尚、アプリケーションプログラムが独自の利用実績出力機能を持っているときは、統一性のある利用実績のフォーマットに変換して課金システム503にファイル転送で送信する機能を有する。

【0131】次に、本発明の実施の形態のアプリケーション・サービスを提供する方法につき、図12〜図14を参照して、簡単に説明する。図12はユーザがアプリケーションサーバ1 (300)に格納されるアプリケーションプログラムを利用した場合の流れ図である。

【ステップD1]ユーザ端末100は、アプリケーションサーバ1 (300)に接続する。

【ステップD2]アプリケーションサーバ1 (300)は、ユーザ端末100にログイン画面を送信する。

【ステップD3]ユーザ端末100は、アプリケーションサーバ1 (300)にユーザIDとパスワードとを送信する。

【ステップD4]アプリケーションサーバ1 (300)は、ユーザIDとパスワードと必要に応じてユーザ端末100の端末識別子をチェックし、正当と判断すれば、ユーザが利用契約したアプリケーションプログラムの一覧をメニューとして送信する。

【ステップD5]ユーザ端末100は、アプリケーションプログラムを選択するとURIに従ってアプリケーションサーバ1 (300)に接続する。

【ステップD6]アプリケーションサーバ1 (300)は、URIに従って選択されたアプリケーションプログラムを起動する。

【ステップD7]ユーザは、ユーザ端末100でアプリケーションプログラムを利用し、終了する。

【ステップD8]ユーザが、アプリケーションサーバ1 (300)でアプリケーションプログラムが終了すると、利用実績を蓄積する。

【0132】図13はユーザがアプリケーションサーバ2 (400)に格納されるアプリケーションプログラムを利用した場合の流れ図である。

【ステップE1〜E4]ステップD1〜D4と同じである。

【ステップE5]ユーザ端末100は、アプリケーションプログラムを選択するとURIに従ってアプリケーションサーバ2 (400)に接続する。

【ステップE6]アプリケーションサーバ2 (400)は、URIに従って選択されたアプリケーションプログラムを起動する。

【ステップE7]ユーザは、ユーザ端末100でアプリケ

ーションプログラムを利用し、終了する。

【ステップE8]ユーザが、アプリケーションサーバ2 (400)でアプリケーションプログラムが終了すると、利用実績を蓄積する。

【0133】図14は課金の流れ図である。

【ステップF1]サーバ500は、定期的に蓄積した利用実績をユーザ別アプリケーション別に集計する。

【ステップF2]サーバ500は、課金処理を実行して請求書を発行し、ユーザに電子メール、郵便等で送付する。

【ステップF3]ユーザは、利用料金をサーバ500に電子マネー、クレジットカード、口座振込等で支払う。【ステップF4]サーバ500は、ベンダー別アプリケーション利用料金を集計する。

【ステップF5]サーバ500は、ベンダー端末700にアプリケーション利用料金を電子マネー、口座振込等で支払う。

【0134】

【発明の効果】本発明のアプリケーション・サービスを提供する方法及びシステムによれば、認証API及び課金APIを組み込んだアプリケーションプログラムと認証システムとをアプリケーションサーバに格納するだけで、ベンダーはアプリケーション・サービスを提供することができる。即ち、ベンダーがアプリケーション・サービス・プロバイダの事業に容易に参入できるという効果がある。

【0135】また、ベンダーはユーザ認証及び課金処理プログラムを開発する必要がなくなるので、新たなアプリケーションプログラムを開発することができる。この場合、この新たなアプリケーションプログラムをアプリケーションサーバに格納するだけで、ベンダーは複数のアプリケーション・サービスを提供することができる。即ち、ユーザが利用可能なアプリケーションプログラムの数若しくは種類が多くなるという効果がある。ここで、複数のアプリケーションプログラムがそれぞれ認証APIを介して一の認証システムによって、ユーザを識別することができる。換言すれば、複数のアプリケーションプログラム間でユーザ管理の統一性を実現することができる。

【0136】更に、別のベンダーが別のアプリケーションプログラムを開発した場合、複数のアプリケーションプログラムと一の認証システムとをアプリケーションサーバに別のアプリケーションプログラムを格納するだけで、別のベンダーもアプリケーション・サービスを提供することができる。即ち、アプリケーションサーバは複数のベンダーのアプリケーションプログラムをユーザに有償で利用させることができる。ここで、複数のベンダーのアプリケーションプログラムがそれぞれ課金APIを介して一の課金システムによって、利用実績を集計して利用金額を算出することができる。換言すれば、複

数のベンダーの間で課金管理の統一性を実現することができる。課金システムはユーザー別に利用金額を算出してユーザーに支払いを求める一方、課金システムはベンダー別に利用金額を算出してベンダーに支払う。これにより、ユーザーは利用代金の支払い等の事務手続が煩雑でないという効果がある。

【図面の簡単な説明】

【図1】 実施の形態1のアプリケーション・サービスを提供するための準備をするシステムの構成例を示すブロック図である。

【図2】 実施の形態1のアプリケーションプログラムを準備する方法を説明するためのフローチャート図である。

【図3】 実施の形態1のアプリケーションプログラムを利用するユーザーの登録方法を説明するためのフローチャート図である。

【図4】 実施の形態1のアプリケーション情報表示画面である。

【図5】 実施の形態1のアプリケーション・サービスを提供するシステムの構成例を示すブロック図である。

【図6】 実施の形態1の認証APIをアプリケーションプログラムに組み込む方法を示すフローチャートである。

【図7】 実施の形態1の課金APIをアプリケーションプログラムに組み込む方法を示すフローチャートである。

【図8】 実施の形態1の認証保持システムが保持する例である。

【図9】 実施の形態1のメニュー保持システムが保持する例である。

【図10】 実施の形態1の課金システムが保持する例である。

【図11】 実施の形態2のアプリケーション・サービスを提供するシステムの構成例を示すブロック図であ

＊ する。

【図12】 実施の形態2のアプリケーションサーバ1のアプリケーションプログラムを利用した場合の流れ図である。

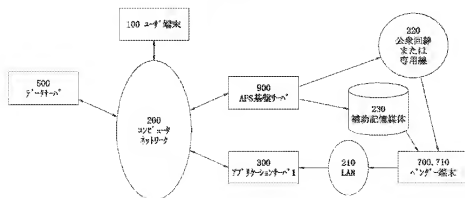
【図13】 実施の形態2のアプリケーションサーバ2のアプリケーションプログラムを利用した場合の流れ図である。

【図14】 実施の形態2の課金の流れ図である。

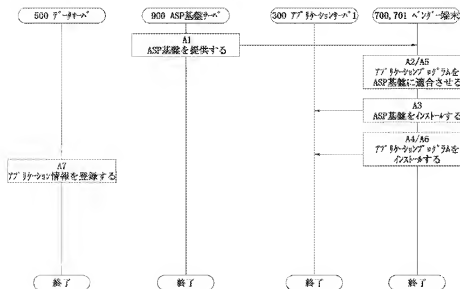
【符号の説明】

100	ユーザ端末
101	Webブラウザ
200	コンピュータネットワーク
210	LAN
220	公衆回線または専用線
230	補助記憶媒体
300, 400	アプリケーションサーバ
301	認証システム
302	動的メニュー生成システム
303	認証API
304	課金API
310, 320	アプリケーションプログラム
390	Webサーバ
500	サーバ
501	認証保持システム
501-1	認証
501-2	認証証明書
502	メニュー保持システム
502-1	アプリケーションプログラムリスト
502-2	サーバリスト
503	課金システム
503-1	単価マスタ
503-2	利用実績
700, 701	ベンダー端末
900	APS基盤サーバ

【図1】



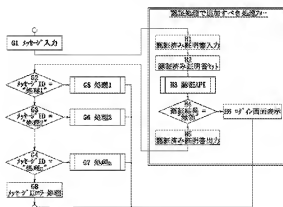
【図2】



【図3】



【図6】



【図4】

各表示ボタンをクリックすると、それぞれ機能詳細説明画面、アプリケーション画面、契約条件画面(料金を含む)が表示される

検索条件 検索ボタン

アプリケーション名称

1. 英日翻訳

機能概要

詳細表示

デモ表示

料金表示

選択

2. 百科事典

機能概要

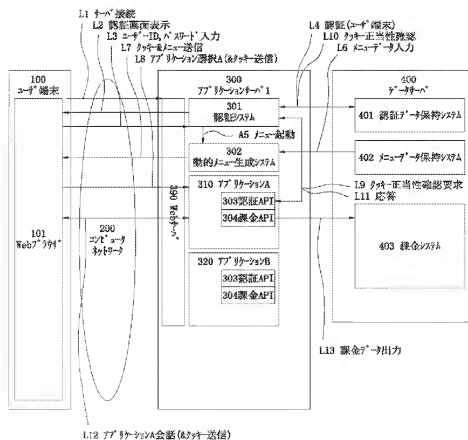
詳細表示

デモ表示

料金表示

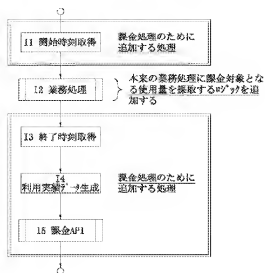
契約申し込みボタン

【図5】



【図7】

(図6の処理1～nの内部の処理フロー)



【図9】

データベース保持システム(502)で管理するデータの一例の実施例

502-1 アプリケーションプログラムリスト

ユーザ ID
アプリケーションプログラム名

502-2 サーバリスト

アプリケーションプログラム名
サーバ 端末識別名
プログラムファイル名

【図8】

認証データ保持システム(501)で管理するデータの一例の実施例

501-1 認証データ

ユーザ ID
パスワード
端末識別子

501-2 認証証明書データ

暗号化した認証証明書	
認証済み証明書	ユーザ ID
認証時刻(年月日時分秒)	

【図10】

課金システム(503)で管理するデータの一例の実施例

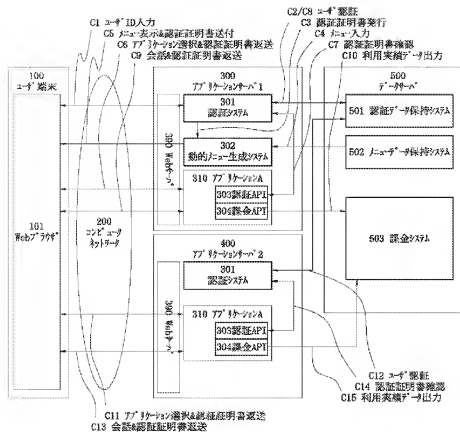
503-1 単価マスタ

ユーザ ID
アプリケーションプログラム名
基本料金
課金項目1の単価
課金項目2の単価
...

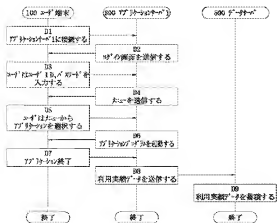
503-2 利用実績データ

ユーザ ID
アプリケーションプログラム名
処理開始時刻
処理時間
課金項目1
課金項目2
...

【図11】



【図12】




```

graph TD
    U100([100 ユーザ端末]) -- E1 --> U100
    U100 -- E3 --> U100
    U300([300 アドブションサーバ1]) -- E2 --> U100
    U100 -- E4 --> U300
    U300 -- E6 --> U400([400 アドブションサーバ2])
    U400 -- E8 --> U500([500 データサーバ])
    U500 -- E9 --> U500
    U100 -- E7 --> E100([終了])
    U300 -- E5 --> E300([終了])
    U400 -- E8 --> E400([終了])
    U500 -- E9 --> E500([終了])
  
```

Figure 1 is a flowchart illustrating the system operation. The process involves four main components: 100 ユーザ端末 (User Terminal), 300 アドブションサーバ1 (Adoption Server 1), 400 アドブションサーバ2 (Adoption Server 2), and 500 データサーバ (Data Server). The flow starts with the User Terminal performing step E1 (login) and step E3 (inputting ID and password). The Adoption Server 1 performs step E2 (sending ID screen) and step E4 (sending ID). The process then moves to the Adoption Server 2, which performs step E6 (login failure) and step E8 (sending usage fee). The Data Server performs step E9 (charging usage fee). The process ends at '終了' (End) for each component.

```

graph TD
    A([100 ユーザ-請求]) --> F3[F3  
利用料金を支払う]
    B([500 データ-提供]) --> P1[P1  
定期的に利用実績データを  
ユーザ別/関与別に集計する]
    P1 --> F2[F2  
課金処理を実行して  
請求書を発行する]
    F2 --> F3
    F2 --> F4[F4  
ペンダ-別/関与別利用料  
を集計する]
    F4 --> F5[F5  
ペンダ-別の利用料金を支払う]
    C([700 ペンダ-請求]) --> F5
    F3 --> D([終了])
    F4 --> E([終了])
    F5 --> F([終了])
  
```